DAY3

Aligning PE reads to a reference genome and BAM refinement

25.11.2015

Chiara Batini
cb334@le.ac.uk

BBASH at

before starting

are you in your scratch directory?

```
pwd
cd $CINECA_SCRATCH
```

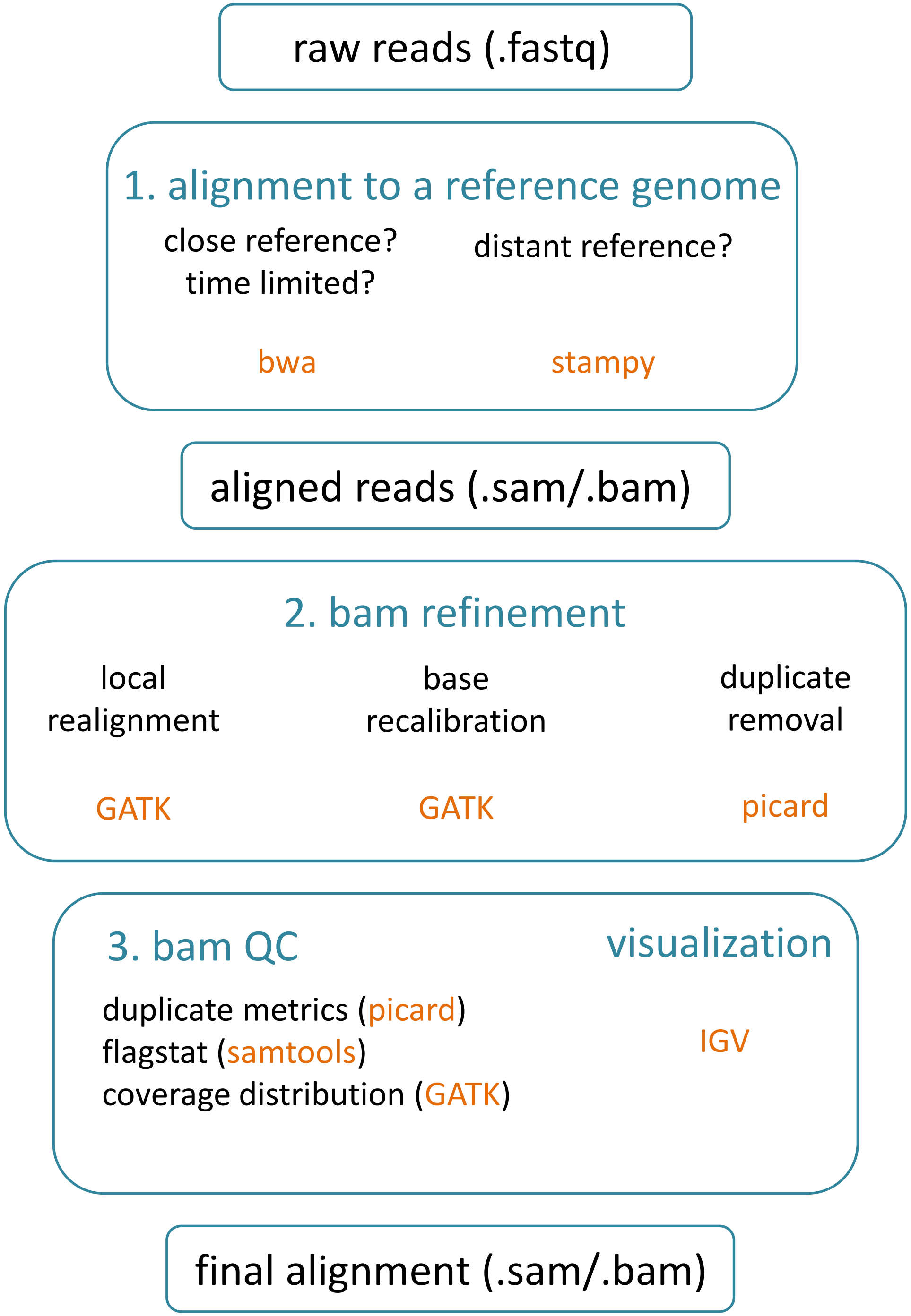
copy folder day3 from teaching directory to yours

```
cp -r /pico/scratch/userexternal/cbatini0/day3/ .
cd day3/

ls
```

The folder VariantCalling contains:

- reads in fastq format (in two folders: lane1 and lane2)
- reference genome in fasta format (Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa)
- coordinates of the yeast mtDNA (mito.intervals)
- the pdf of these slides (day3_mapping_BAM_refinement_nov2015.pdf)
- the handbook for today (day3_mapping_BAM_refinement_handbook_nov2015.pdf)



raw reads (.fastq)

1. alignment to a reference genome

close reference?
time limited?

bwa

distant reference?

stampy

aligned reads (.sam/.bam)

2. bam refinement

local realignment

GATK

base recalibration

GATK

duplicate removal

picard

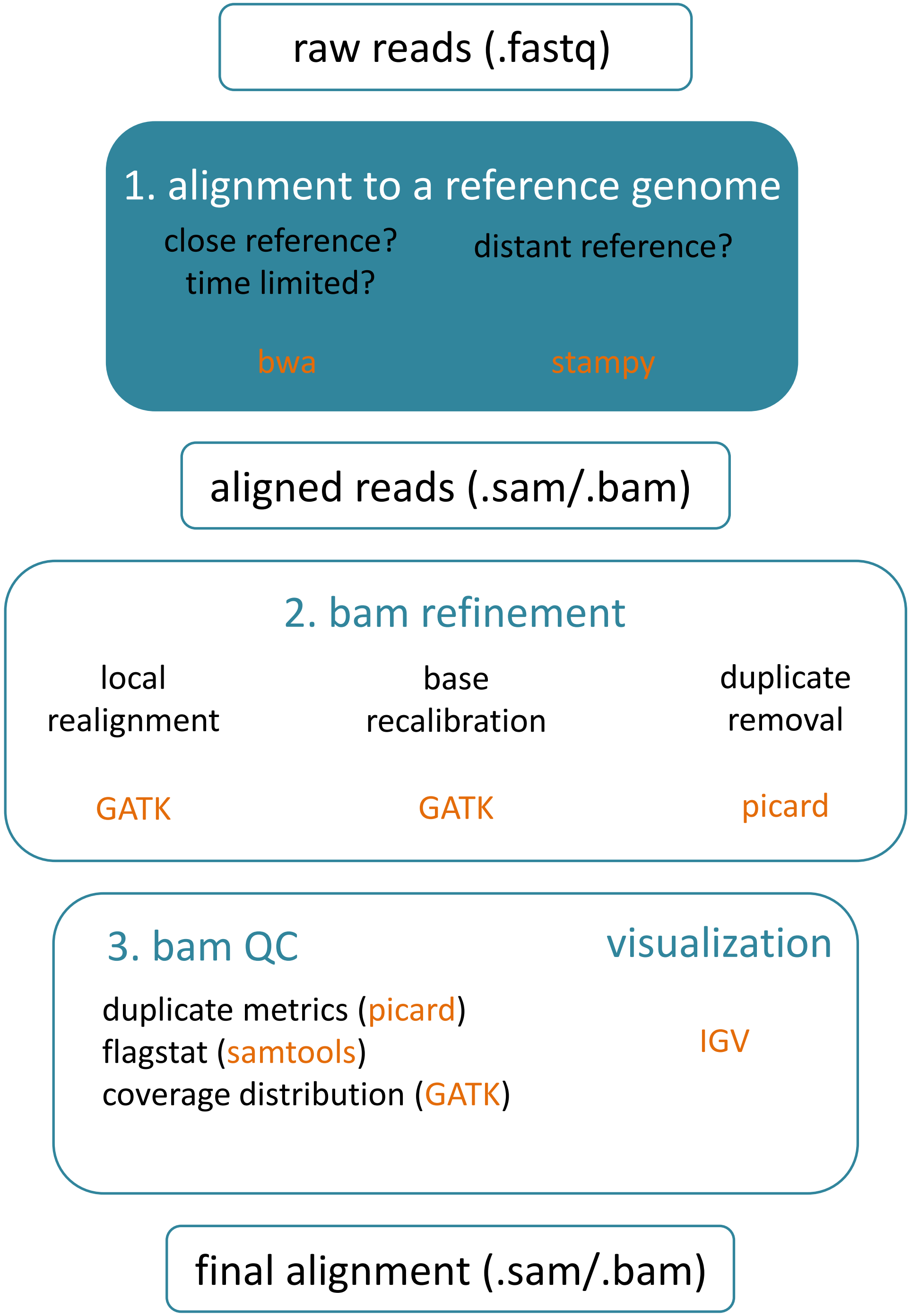
3. bam QC

duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)

visualization

IGV

final alignment (.sam/.bam)



raw reads (.fastq)

1. alignment to a reference genome

close reference?
time limited?

bwa

distant reference?

stampy

aligned reads (.sam/.bam)

2. bam refinement

local
realignment

GATK

base
recalibration

GATK

duplicate
removal

picard

3. bam QC

duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)

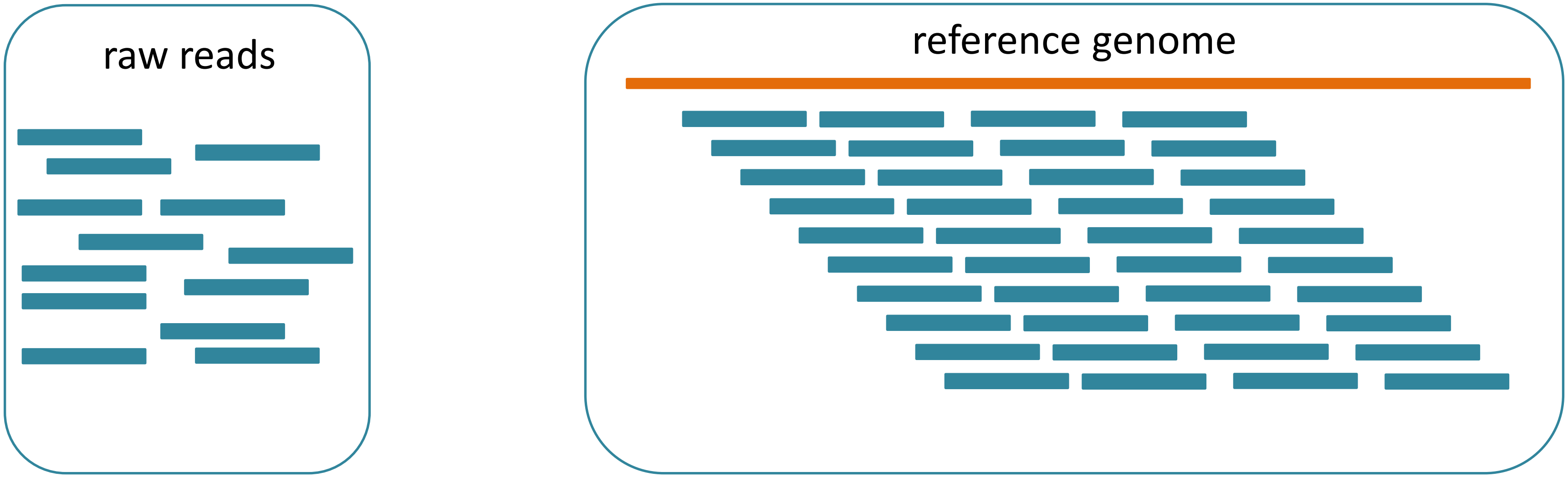
visualization

IGV

final alignment (.sam/.bam)

alignment to a reference genome

alignment – process of determining the most likely location within the genome for the observed DNA read



alignment to a reference genome

short reads: ranging from the initial 36bp of Illumina to the current 1kb of 454/Roche
the shorter the read, the harder is to find its location in the genome

big amount of data: computationally challenging for memory and speed

trade-off: speed vs sensitivity – the higher the accuracy, the longer the alignment run

two classes of methods:

Burrows-Wheeler

- Fast
- less robust at high divergence with reference genome
- e.g. bwa

Hashing

- slow (needs more memory)
- robust at high divergence with reference genome
- e.g. stampy

Further reading: “A survey of sequence alignment algorithms for next-generation sequencing”
Li H. and Homer N. 2010. Briefing In Bioinformatics

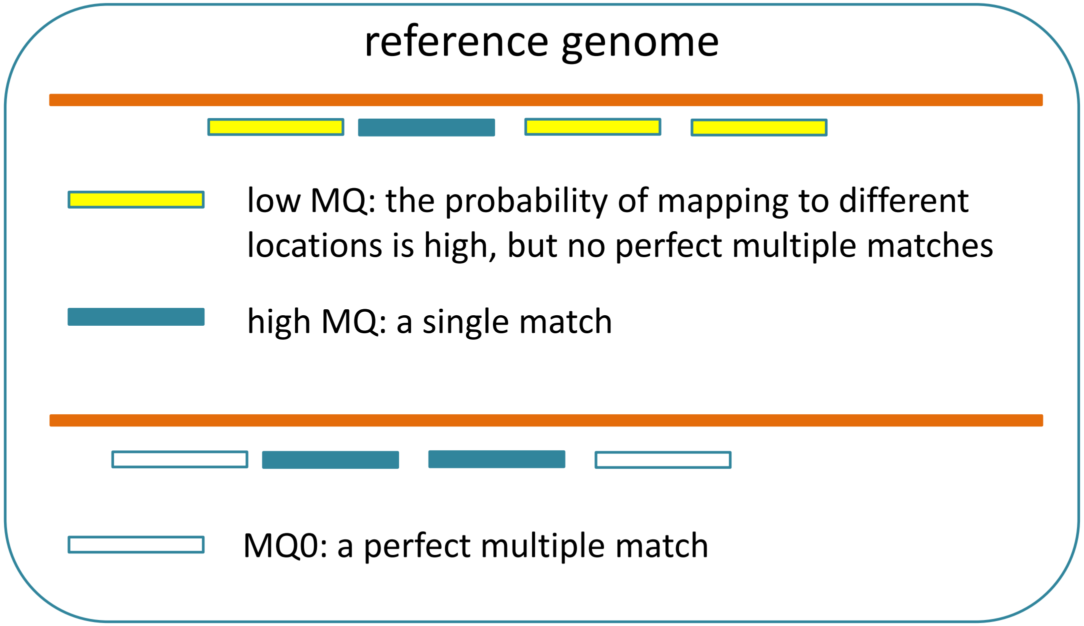
alignment to a reference genome – mapping qualities (MQ)

What if there are several possible places to align your sequencing read?
This may be due to:

- Repeated elements in the genome
- Low complexity sequences
- Reference errors and gaps

MQ is a phred-score of the quality of the alignment
With paired-end reads: mapping quality is determined on the pair, thus even if one read can be mapped in several places, the mapping of its pair can help to locate it properly.

raw reads


reference genome

low MQ: the probability of mapping to different locations is high, but no perfect multiple matches

high MQ: a single match

MQ0: a perfect multiple match

SAM/BAM format

SAM – sequence alignment map
BAM – binary alignment map

Standard formats for alignment
BAM is the binary version of SAM – reduced size, easier to store and to access
but the full information is not readable by human eye

software websites

software	**website**
bwa	http://bio-bwa.sourceforge.net/
picard	http://picard.sourceforge.net/
samtools	http://samtools.sourceforge.net/
GATK	http://www.broadinstitute.org/gatk/
tablet	http://bioinf.scri.ac.uk/tablet/
vcftools	http://vcftools.sourceforge.net/

alignment to a reference genome – details

Characteristics of our experiment:

- Yeast genome: 12.5 Mbp; 16 chromosomes
- Whole genome sequencing
- Paired-end reads, 108bp, one library, 2 lanes

You should be in the right directory, otherwise move there

```
(cd /pico/scratch/userexternal/username/day3)
```

alignment to a reference genome - indexing

create the index of the reference genome (for bwa, samtools and picard)

bwa index

This is a FM-index – specific to the algorithm behind this aligner

```
module load bwa
bwa index -a is
Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
```

index .fai

The index file stores records of sequence identifier, length, the offset of the first sequence character in the file, the number of characters per line, and the number of bytes per line.

```
module load autoload samtools
samtools faidx Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
```

alignment to a reference genome - dictionary

create the dictionary of the reference genome (for samtools, gatk and picard)

dictionary .dict   List of contigs included in the fasta file of the reference genome

```
module load autoload picard
java -jar
/cineca/prod/applications/picard/1.119/binary/bin/CreateSequence
Dictionary.jar R=Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
O=Saccharomyces_cerevisiae.EF4.68.dna.toplevel.dict
```

keep index and dictionary files in the same directory of the reference file

alignment to a reference genome – mapping with bwa mem

From bwa website:
“BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. **It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM.** The first algorithm is designed for Illumina sequence reads up to 100bp, while the rest two for longer sequences ranged from 70bp to 1Mbp. **BWA-MEM and BWA-SW share similar features such as long-read support and split alignment, but BWA-MEM, which is the latest, is generally recommended for high-quality queries as it is faster and more accurate.** BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.”

paired-end alignment (per lane)

It uses the reference genome and the reads to create a SAM file

```
bwa mem -M Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
lane1/s-7-1.fastq lane1/s-7-2.fastq > lane1.sam
```

The option –M marks shorter split hits as secondary, and not supplementary (for Picard/GATK compatibility). It changes the flag, so that “old” tools can manage the bam file. A split read is a read which split maps to two different far apart (same or diff chromosome), a chimeric read. This can happen for a read pair too.

alignment to a reference genome – from sam to bam with samtools

sam-to-bam

```
samtools view -S -b lane1.sam -o lane1.bam
```

sort the bam (this adds the bam extension automatically!)
It sorts alignments by leftmost coordinates

```
samtools sort lane1.bam lane1_sorted
```

index the bam

```
samtools index lane1_sorted.bam
```

Can you guess the extension of this file? Check it in your folder... (use unix `ls` and options)

Can you now repeat this process (paired-end alignment with bwa plus conversion to bam and sorting and indexing) on lane 2?

SAM/BAM format

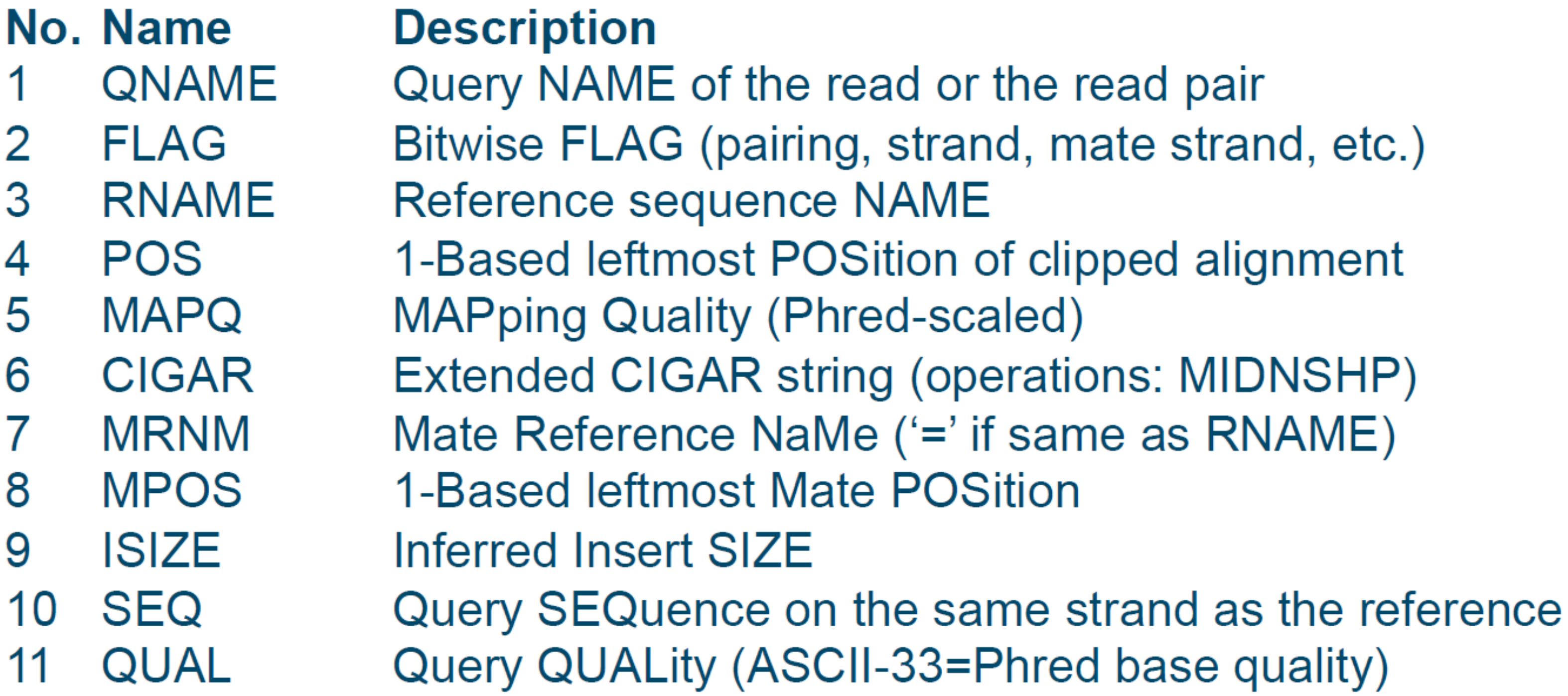
SAM – sequence alignment map
BAM – binary alignment map

They consist of two parts:
Header – contains information about the sample
Alignment – contains location and qualities for all the reads

Header contains:
@HD – header line; format version
@SQ – Reference sequence dictionary; one per chromosome
@RG – Read group
@PG – Program
@CO – comment

SAM/BAM format

Alignment contains one line per read, and each line contains 12 columns:



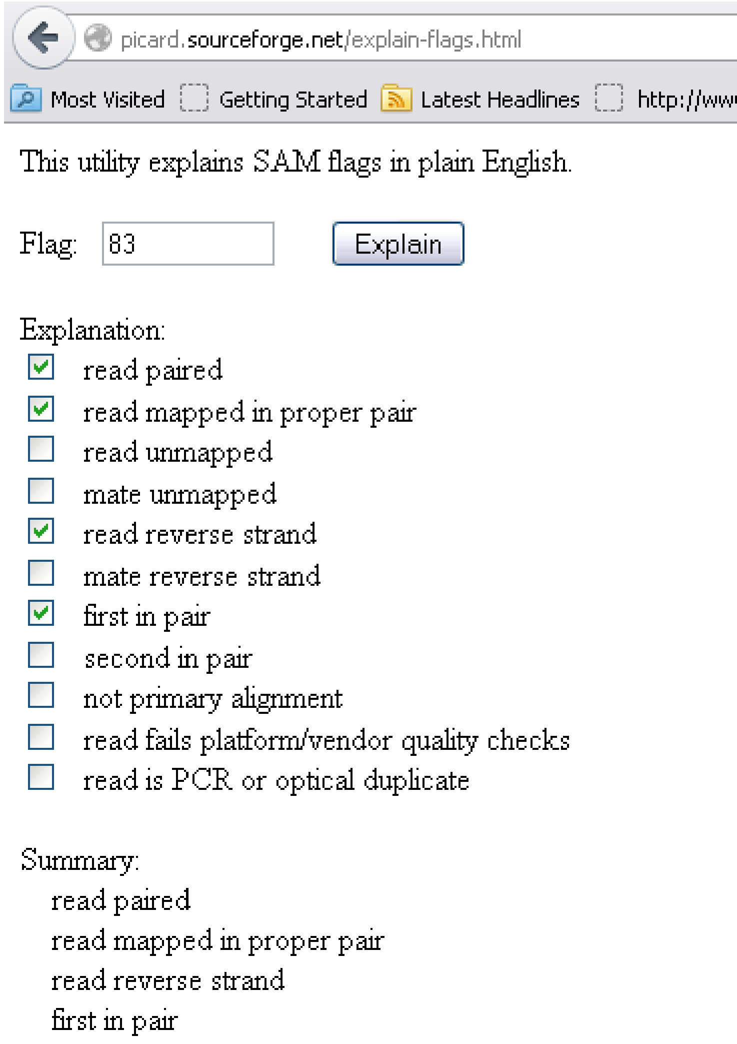
No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe (‘=’ if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

bitwise FLAG

It is an integer, but it represents the sum of different values.

You can find a detailed explanation in the sam/bam format specification (http://samtools.sourceforge.net/SAMv1.pdf).

However there is a tool online which provides a quick “translation” (https://broadinstitute.github.io/picard/explain-flags.html)



This utility explains SAM flags in plain English.

Flag: 83 Explain

Explanation:

- ☑ read paired
- ☑ read mapped in proper pair
- □ read unmapped
- □ mate unmapped
- ☑ read reverse strand
- □ mate reverse strand
- ☑ first in pair
- □ second in pair
- □ not primary alignment
- □ read fails platform/vendor quality checks
- □ read is PCR or optical duplicate

Summary:

read paired
read mapped in proper pair
read reverse strand
first in pair

CIGAR string

It is a compact representation of sequence alignment. It includes:

- M – match or mismatch
- I – insertion
- D – deletion

SAM extends these to include a few others (check http://samtools.sourceforge.net/SAMv1.pdf)

read:	**ACGCA–TGCAGT**
ref:	**ACTCAGTG ––GT**
cigar	**5M1D2M2I2M**

So, what is the cigar line of…?

read:	**ACGTCATG –––CAGT**
ref:	**ACG–CATGCGGCAGT**
cigar	

CIGAR string

It is a compact representation of sequence alignment. It includes:

- M – match or mismatch
- I – insertion
- D – deletion

SAM extends these to include a few others (check http://samtools.sourceforge.net/SAMv1.pdf)

read:	**ACGCA–TGCAGT**
ref:	**ACTCAGTG ––GT**
cigar	**5M1D2M2I2M**

So, what is the cigar line of…?

read:	**ACGTCATG –––CAGT**
ref:	**ACG–CATGCGGCAGT**
cigar	**3M1I4M3D4M**

SAM/BAM format

```
@HD	VN:1.0	SO:coordinate
@SQ	SN:chr1	LN:249250621	UR:file:/home/chiara	M5:1b22b98cdeb4a9304cb5d48026a85128
@SQ	SN:chr2	LN:243199373	UR:file:/home/chiara	M5:a0d9851da00400dec1098a9255ac712e
@SQ	SN:chr3	LN:198022430	UR:file:/home/chiara	M5:641e4338fa8d52a5b781bd2a2c08d3c3
@SQ	SN:chr4	LN:191154276	UR:file:/home/chiara	M5:23dccd106897542ad87d2765d28a19a1
@SQ	SN:chr5	LN:180915260	UR:file:/home/chiara	M5:0740173db9ffd264d728f32784845cd7
@SQ	SN:chr6	LN:171115067	UR:file:/home/chiara	M5:1d3a93a248d92a729ee764823acbbc6b
@SQ	SN:chr7	LN:159138663	UR:file:/home/chiara	M5:618366e953d6aaad97dbe4777c29375e
@SQ	SN:chr8	LN:146364022	UR:file:/home/chiara	M5:96f514a9929e410c6651697bded59aec
@SQ	SN:chr9	LN:141213431	UR:file:/home/chiara	M5:3e273117f15e0a400f01055d9f393768
@SQ	SN:chr10	LN:135534747	UR:file:/home/chiara	M5:988c28e000e84c26d552359af1ea2e1d
@SQ	SN:chr11	LN:135006516	UR:file:/home/chiara	M5:98c59049a2df285c76ffb1c6db8f8b96
@SQ	SN:chr12	LN:133851895	UR:file:/home/chiara	M5:51851ac0e1a115847ad36449b0015864
@SQ	SN:chr13	LN:115169878	UR:file:/home/chiara	M5:283f8d7892baa81b510a015719ca7b0b
@SQ	SN:chr14	LN:107349540	UR:file:/home/chiara	M5:98f3cae32b2a2e9524bc19813927542e
@SQ	SN:chr15	LN:102531392	UR:file:/home/chiara	M5:e5645a794a8238215b2cd77acb95a078
@SQ	SN:chr16	LN:90354753	UR:file:/home/chiara	M5:fc9b1a7b42b97a864f56b348b06095e6
@SQ	SN:chr17	LN:81195210	UR:file:/home/chiara	M5:351f64d4f4f9ddd45b35336ad97aa6de
@SQ	SN:chr18	LN:78077248	UR:file:/home/chiara	M5:b15d4b2d29dde9d3e4f93d1d0f2cbc9c
@SQ	SN:chr19	LN:59128983	UR:file:/home/chiara	M5:1aacd71f30db8e561810913e0b72636d
@SQ	SN:chr20	LN:63025520	UR:file:/home/chiara	M5:0dec9660ec1efaaf33281c0d5ea2560f
@SQ	SN:chr21	LN:48129895	UR:file:/home/chiara	M5:2979a6085bfe28e3ad6f552f361ed74d
@SQ	SN:chr22	LN:51304566	UR:file:/home/chiara	M5:a718acaa6135fdca8357d5bfe94211dd
@SQ	SN:chrX	LN:155270560	UR:file:/home/chiara	M5:7e0e2e580297b7764e31dbc80c2540dd
@SQ	SN:chrY	LN:59373566	UR:file:/home/chiara	M5:1e86411d73e6f00a10590f976be01623
@SQ	SN:chrM	LN:16571	UR:file:/home/chiara	M5:d2ed829b8a1628d16cbeee88e88e39eb
@RG	ID:1	PL:illumina	PU:1	LB:1	SM:003_stampy_automasked
@PG	ID:GATK IndelRealigner	VN:1.2-62-g41ddc7b
@PG	ID:stampy	VN:1.0.13_(r1160)
@PG	ID:GATK TableRecalibration	VN:1.2-62-g41ddc7b
@CO	TM:Sat, 31 Dec 2011 10:59:43 GMT
HWI-ST427:142:D08WKACXX:6:1202:4868:142425	163	chr1	10002	37	101M	=	10075	173
	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCC
	>B<@@B>C<@?>:B>@@BAB@??B<A?9?@@A8>65<B<;?B@@:??B=B@=@?@C9?@90@2?>?@A9=?;*+9*);*0<>AA>C9?@>AC#########
```

- Which programs have been run on this sam?
- Is the read mapped in a proper pair?
- Does the read contain any indel compared to the reference genome?
- Is the pair on the same chromosome? What is the insert size of the pair?

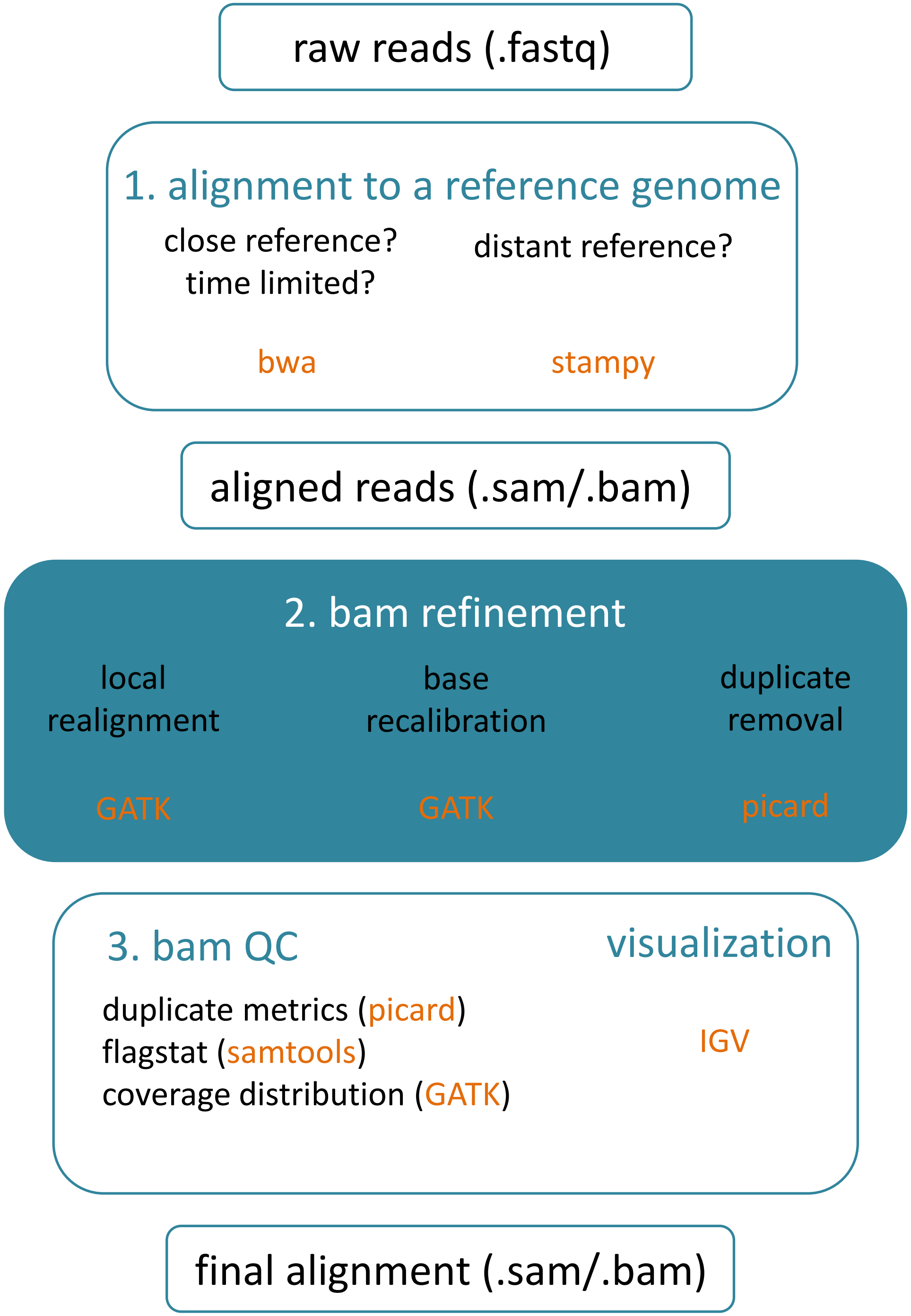
BAM visualization – check the header of your BAM

use samtools to check the header of the BAM

```
samtools view -H lane1_sorted.bam
```

How many chromosomes are present in your header?
Which version of the SAM is it?

use unix command `more` on your SAM file and check what is after the header...



raw reads (.fastq)

1. alignment to a reference genome

close reference?
time limited?

bwa

distant reference?

stampy

aligned reads (.sam/.bam)

2. bam refinement

local realignment

GATK

base recalibration

GATK

duplicate removal

picard

3. bam QC

duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)

visualization

IGV

final alignment (.sam/.bam)

BAM refinement

Input: BAM

Three main steps:

1. Local realignment
2. Base quality recalibration
3. Duplicate removal

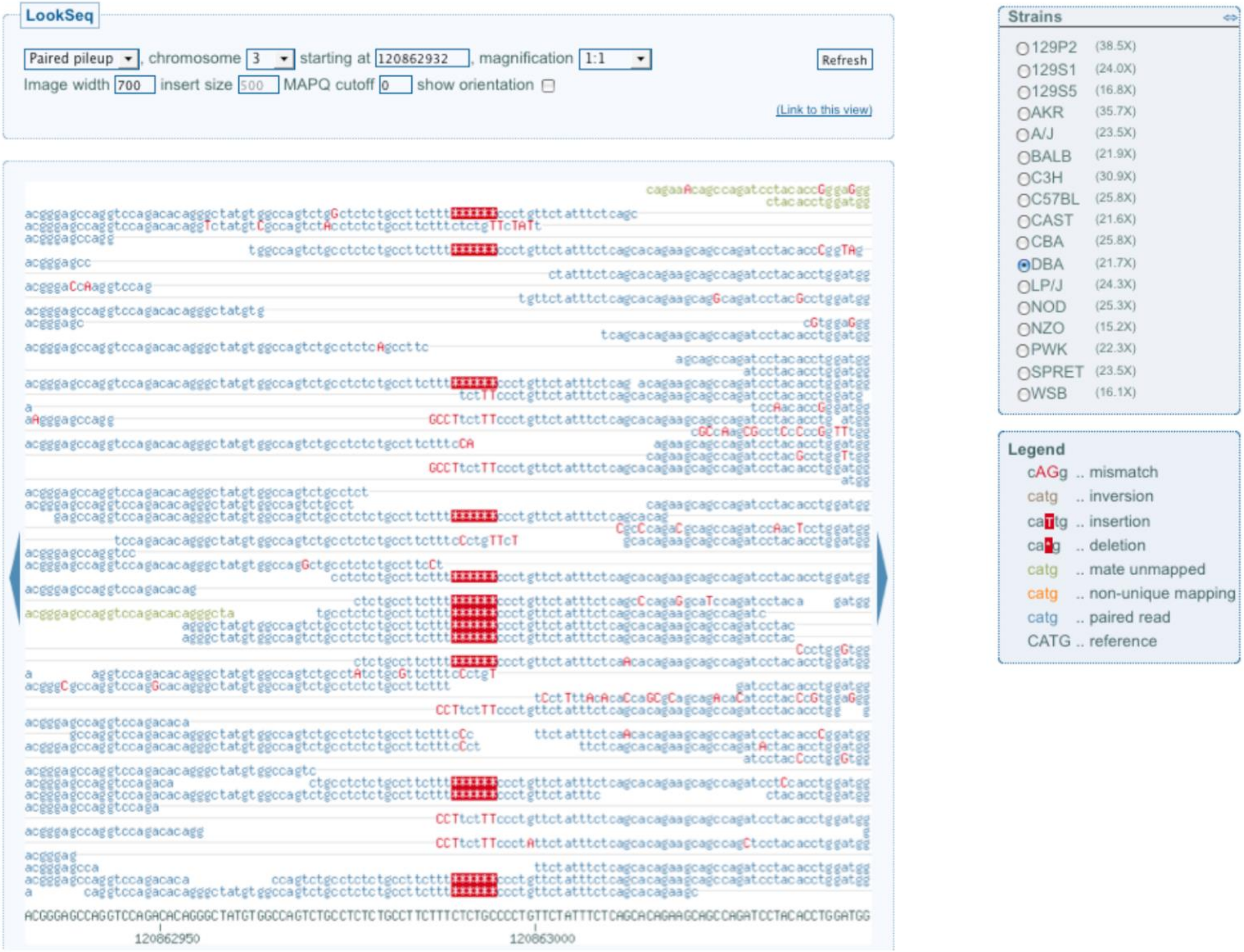
Output: BAM

Good practice:

Run step 1 and 2 at lane level, while step 3 must be run at library level

BAM refinement – local realignment

Short indels in the sample relative to the reference sequence can pose difficulties for alignment programs. Indels occuring towards the ends of the reads are often not aligned correctly, introducing an excess of SNPs



LookSeq

Paired pileup, chromosome 3, starting at 120862932, magnification 1:1 Refresh

Image width 700 insert size 500 MAPQ cutoff 0 show orientation □

(Link to this view)

```
ACGGGAGCCAGGTCCAGACACAGGGCTATGTGGCCAGTCTGCCTCTCTGCCTTCTTTCTCTGCCCCTGTTCTATTTCTCAGCACAGAAGCAGCCAGATCCTACACCTGGATGG
         120862950                                        120863000
```

Strains

○129P2	(38.5X)
○129S1	(24.0X)
○129S5	(16.8X)
○AKR	(35.7X)
○A/J	(23.5X)
○BALB	(21.9X)
○C3H	(30.9X)
○C57BL	(25.8X)
○CAST	(21.6X)
○CBA	(25.8X)
●DBA	(21.7X)
○LP/J	(24.3X)
○NOD	(25.3X)
○NZO	(15.2X)
○PWK	(22.3X)
○SPRET	(23.5X)
○WSB	(16.1X)

Legend

cAGg .. mismatch
catg .. inversion
caTtg .. insertion
ca*g .. deletion
catg .. mate unmapped
catg .. non-unique mapping
catg .. paired read
CATG .. reference

BAM refinement – GATK local realignment

It uses the full alignment context to determine whether the indel exists.

Two-step process:

1. RealignerTargetCreator: it determines the small suspicious intervals which are likely in need of realignment
2. IndelRealigner: it runs the realignment on those intervals

notes:

- having a list of known indels helps
- it doesn’t work on 454 reads or from similar technologies (as from GATK webpage)
- however I have used on IonTorrent data and it worked fine…

BAM refinement – before starting

merge BAMs per library

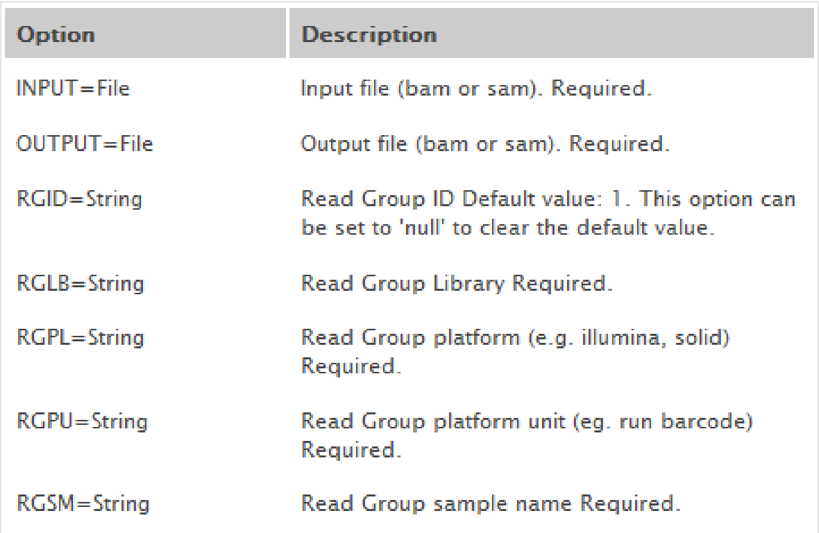
```
java -jar
/cineca/prod/applications/picard/1.119/binary/bin/MergeSamFiles
.jar INPUT=lane1_sorted.bam INPUT=lane2_sorted.bam
OUTPUT=library.bam
```

GATK wants read groups to be present, and it complains if they are not there. Do we have RG? Check the header…

BAM refinement – before starting

In this case we need to add a read group for the library with picard
(please keep in mind that there is a way to do this during the alignment with bwa!)

```
java -jar
/cineca/prod/applications/picard/1.119/binary/bin/AddOrReplace
ReadGroups.jar INPUT=library.bam OUTPUT=library_RG.bam RGID=1
RGLB=library RGPL=Illumina RGPU=lane1_2 RGSM=yeast
```



Option	Description
INPUT=File	Input file (bam or sam). Required.
OUTPUT=File	Output file (bam or sam). Required.
RGID=String	Read Group ID Default value: 1. This option can be set to 'null' to clear the default value.
RGLB=String	Read Group Library Required.
RGPL=String	Read Group platform (e.g. illumina, solid) Required.
RGPU=String	Read Group platform unit (eg. run barcode) Required.
RGSM=String	Read Group sample name Required.

sort and index the library BAM file with samtools

BAM refinement – local realignment with GATK

```
module load autoload gatk/3.3.0
```

1. RealignerTargetCreator:

```
java -jar /cineca/prod/applications/gatk/3.3.0/jre--
1.7.0_72/GenomeAnalysisTK.jar -I library_RG_sorted.bam
-R Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
-T RealignerTargetCreator
-o library_targets.intervals
```

2. IndelRealigner:

```
java -jar /cineca/prod/applications/gatk/3.3.0/jre--
1.7.0_72/GenomeAnalysisTK.jar -I library_RG_sorted.bam
-R Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
-T IndelRealigner
-targetIntervals library_targets.intervals
-o library_RG_sorted_lr.bam
```

BAM refinement – base quality recalibration

Each base call has an associated base call quality (phred-scale).
Rule of thumb: anything less than Q20 is not useful data.

The quality of a call depends on multiple factors (e.g. position in the read, sequence context). In addition, the alignment can provide useful information. Mismatches to the reference are considered errors (unless they are described polymoprhisms).

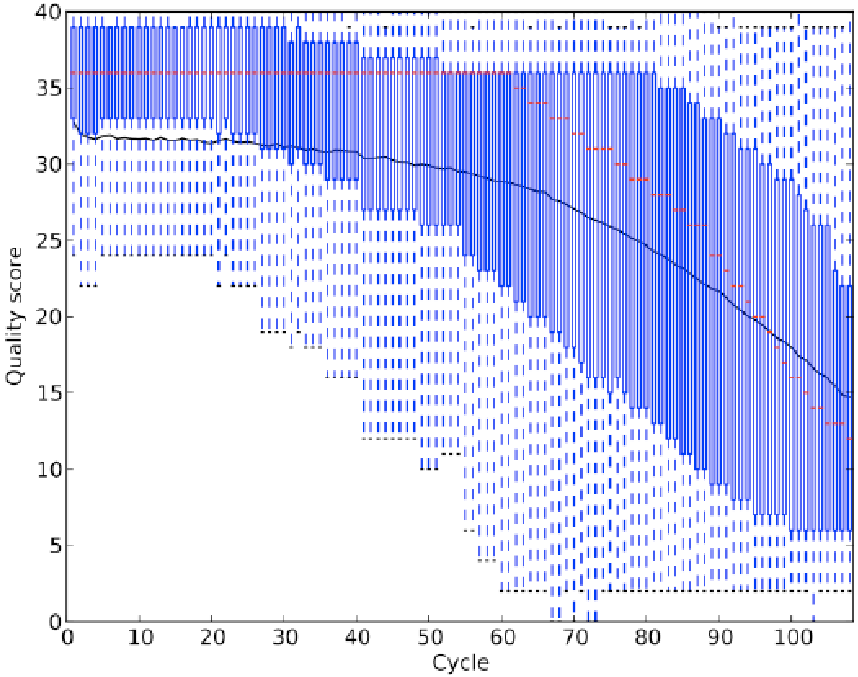
It supports several platforms: Illumina, SOLiD, 454, Complete Genomics, Pacific Biosciences (stated on the website) and IonTorrent (stated in the GATK forum).

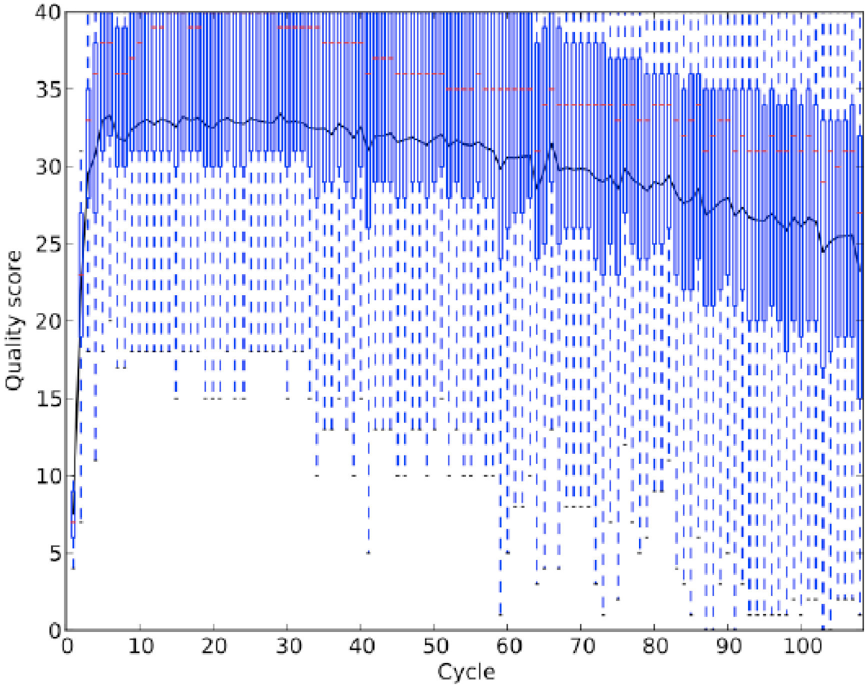
It combines all the available information to re-evaluate the probability of a wrong call at each position in each read.

It requires a catalogue of variable sites!

We will not run it but you can find how to do it at
http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_bqsr_BaseRecalibrator.html

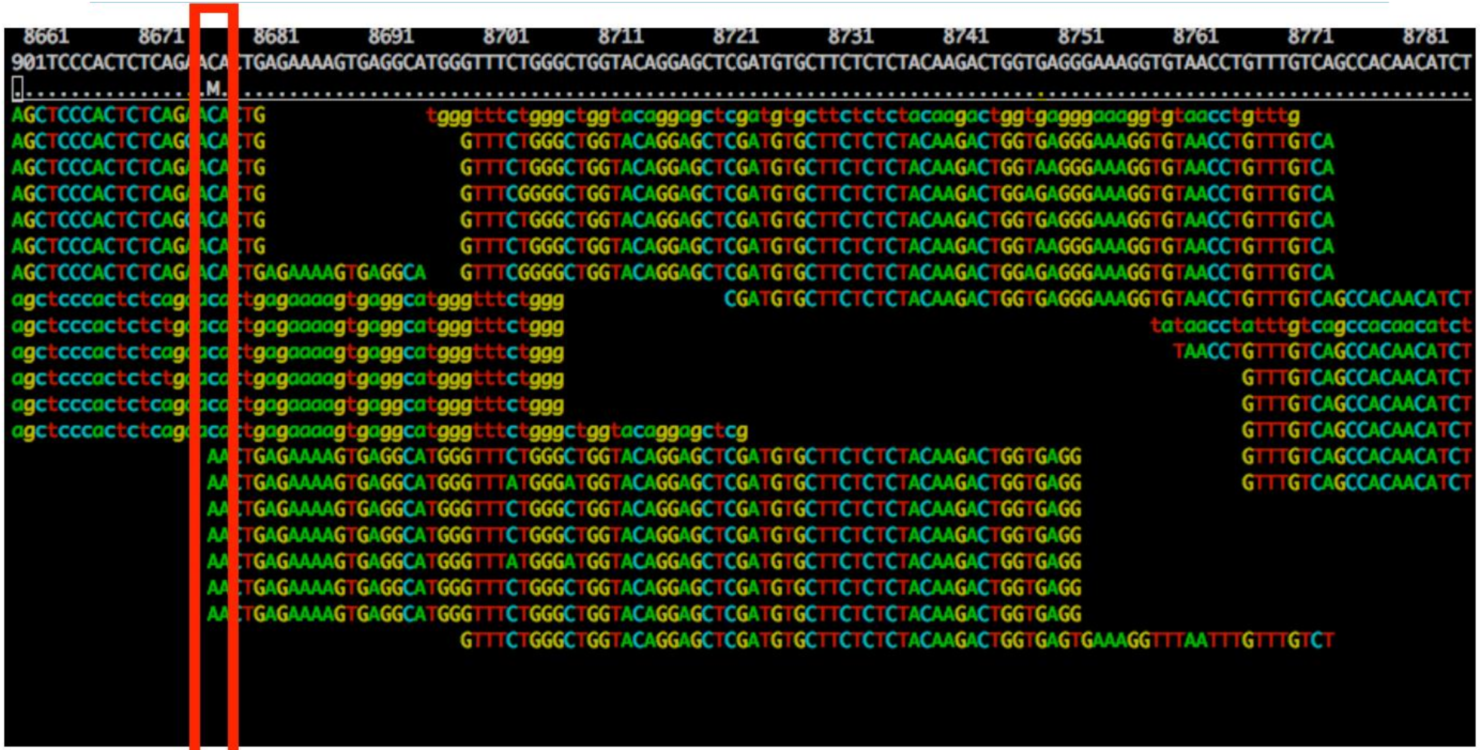
BAM refinement – base quality recalibration





BAM refinement – duplicate removal

PCR is used during library preparation. This can results in duplicate DNA fragments in the final library prep. PCR-free protocols exist but require a large amount of DNA.



```
 8661      8671      8681      8691      8701      8711      8721      8731      8741      8751      8761      8771      8781
901TCCCACTCTCAGAACACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
□................M.............................................................................................................
AGCTCCCACTCTCAGAACACTG              tgggtttctgggctggtacaggagctcgatgtgcttctctctacaagactggtgagggaaaggtgtaacctgtttg
AGCTCCCACTCTCAGCACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGCACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAGAACACTGAGAAAAGTGAGGCA   GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
agctcccactctcagcacactgagaaaagtgaggcatgggtttctggg              CGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
agctcccactctctgcacactgagaaaagtgaggcatgggtttctggg                                                   tataacctatttgtcagccacaacatct
agctcccactctcagcacactgagaaaagtgaggcatgggtttctggg                                                     TAACCTGTTTGTCAGCCACAACATCT
agctcccactctctgcacactgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcagcacactgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcagcacactgagaaaagtgaggcatgggtttctgggctggtacaggagctcg                                           GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AACTGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                                       GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGTGAAAGGTTTAATTTGTTTGTCT
```

NA12005 - chr20:8660-8790

It can result in false SNPs calls. Duplicates may fake a high coverage thus giving high support to some variants.

BAM refinement – duplicate removal

Number of duplicates varies according to the complexity of the library: whole genome experiments show lower percentages of duplicates (<5%) than custom enrichment ones (<30%).

It must be done after alignment and at the library level.

It identifies read-pairs where the outer ends map to the same position on the genome and removes all but one copy.

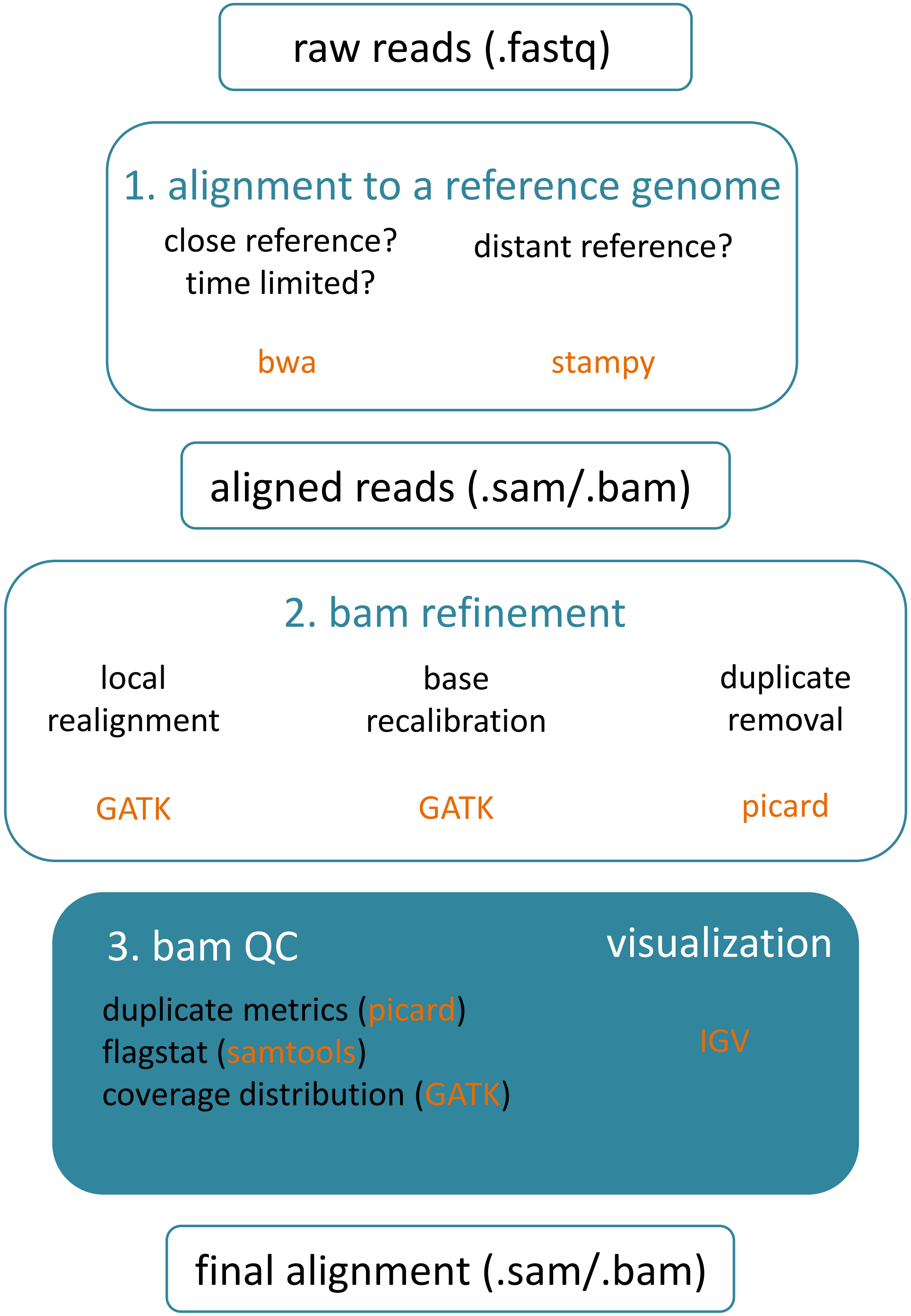
What would you expect in an amplicon-seq experiment?

BAM refinement – duplicate removal

duplicate removal

```
java -jar
/cineca/prod/applications/picard/1.119/binary/bin/Mark
Duplicates.jar INPUT=library_RG_sorted_lr.bam
OUTPUT=library_final.bam METRICS_FILE=dupl_metrics.txt
```

sort and index the final BAM with samtools



raw reads (.fastq)

1. alignment to a reference genome

close reference?
time limited?

bwa

distant reference?

stampy

aligned reads (.sam/.bam)

2. bam refinement

local realignment

GATK

base recalibration

GATK

duplicate removal

picard

3. bam QC

duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)

visualization

IGV

final alignment (.sam/.bam)

BAM QC

How many duplicates do I have? Is that reasonable for my experiment?

How many of my reads mapped back to the reference? How many of these are paired in mapping? How many pairs are mapped to different chromosomes?

How much average coverage do I have? Is the coverage evenly distributed along my region?

BAM QC – picard duplicate metrics

```
java -Xmx2g -jar /cm/shared/apps/picard/1.93/MarkDuplicates.jar
INPUT=library_RG_sorted_lr.bam OUTPUT=library_final.bam
METRICS_FILE=dupl_metrics.txt
```

```
gedit dupl_metrics.txt &
```

```
## net.sf.picard.metrics.StringHeader
# net.sf.picard.sam.MarkDuplicates INPUT=[library_RG_sorted_lr.bam] OUTPUT=library_final.bam METRICS_FILE=duple_metrics.txt
PROGRAM_RECORD_ID=MarkDuplicates PROGRAM_GROUP_NAME=MarkDuplicates REMOVE_DUPLICATES=false ASSUME_SORTED=false
MAX_SEQUENCES_FOR_DISK_READ_ENDS_MAP=50000 MAX_FILE_HANDLES_FOR_READ_ENDS_MAP=8000 SORTING_COLLECTION_SIZE_RATIO=0.25
READ_NAME_REGEX=[a-zA-Z0-9]+:[0-9]:([0-9]+):([0-9]+):([0-9]+).* OPTICAL_DUPLICATE_PIXEL_DISTANCE=100 VERBOSITY=INFO QUIET=false
VALIDATION_STRINGENCY=STRICT COMPRESSION_LEVEL=5 MAX_RECORDS_IN_RAM=500000 CREATE_INDEX=false CREATE_MD5_FILE=false
## net.sf.picard.metrics.StringHeader
# Started on: Tue Jan 06 11:00:13 GMT 2015

## METRICS CLASS        net.sf.picard.sam.DuplicationMetrics
LIBRARY         UNPAIRED_READS_EXAMINED  READ_PAIRS_EXAMINED         UNMAPPED_READS      UNPAIRED_READ_DUPLICATES
        READ_PAIR_DUPLICATES        READ_PAIR_OPTICAL_DUPLICATES        PERCENT_DUPLICATIONESTIMATED_LIBRARY_SIZE
library 17741   233562 150769 3152  4668    1187    0.025756        7678471

## HISTOGRAM java.lang.Double
BIN     VALUE
1.0     1.005031
2.0     1.979951
3.0     2.925663
4.0     3.843042
5.0     4.732936
6.0     5.596169
7.0     6.433539
8.0     7.245822
9.0     8.03377
10.0    8.79811
```

It estimates the return on investment (ROI) for sequencing a library to a higher coverage than the observed coverage.
As one increases the amount of sequencing for a library, the ROI for additional sequencing diminishes because more and more of the reads are duplicates.

BAM QC – samtools flagstat

```
samtools flagstat library_final_sorted.bam > library_flagstat.txt
```

```
640134 + 0 in total (QC-passed reads + QC-failed reads)
4500 + 0 secondary
0 + 0 supplementary
12488 + 0 duplicates
489365 + 0 mapped (76.45%:-nan%)
635634 + 0 paired in sequencing
317817 + 0 read1
317817 + 0 read2
452190 + 0 properly paired (71.14%:-nan%)
467124 + 0 with itself and mate mapped
17741 + 0 singletons (2.79%:-nan%)
7016 + 0 with mate mapped to a different chr
3557 + 0 with mate mapped to a different chr (mapQ>=5)
```

QC: platform/vendor quality check
Duplicates: marked as duplicates by picard
Paired in sequencing: but not necessarily in mapping
Properly paired: it is compatible with the expected insert size
With itself and mate mapped: both are mapped
Singletons: only one is mapped
With mate mapped to a different chromosome: …..

Run flagstat on the BAM file before BAM refinement, can you see any difference?

BAM QC - coverage

coverage per position GATK

```
java -jar /cineca/prod/applications/gatk/3.3.0/jre--
1.7.0_72/GenomeAnalysisTK.jar -T DepthOfCoverage
-R Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa
-I library_final_sorted.bam
-o mito_coverage
-L mito.intervals
```

average coverage

```
gedit mito_coverage.sample_summary &
```

```
sample_id       total   mean    granular_third_quartile   granular_median granular_first_quartile    %_bases_above_15
yeast   3094652         36.08   51      39      27        82.7
Total   3094652         36.08   N/A     N/A     N/A
```

BAM QC - coverage

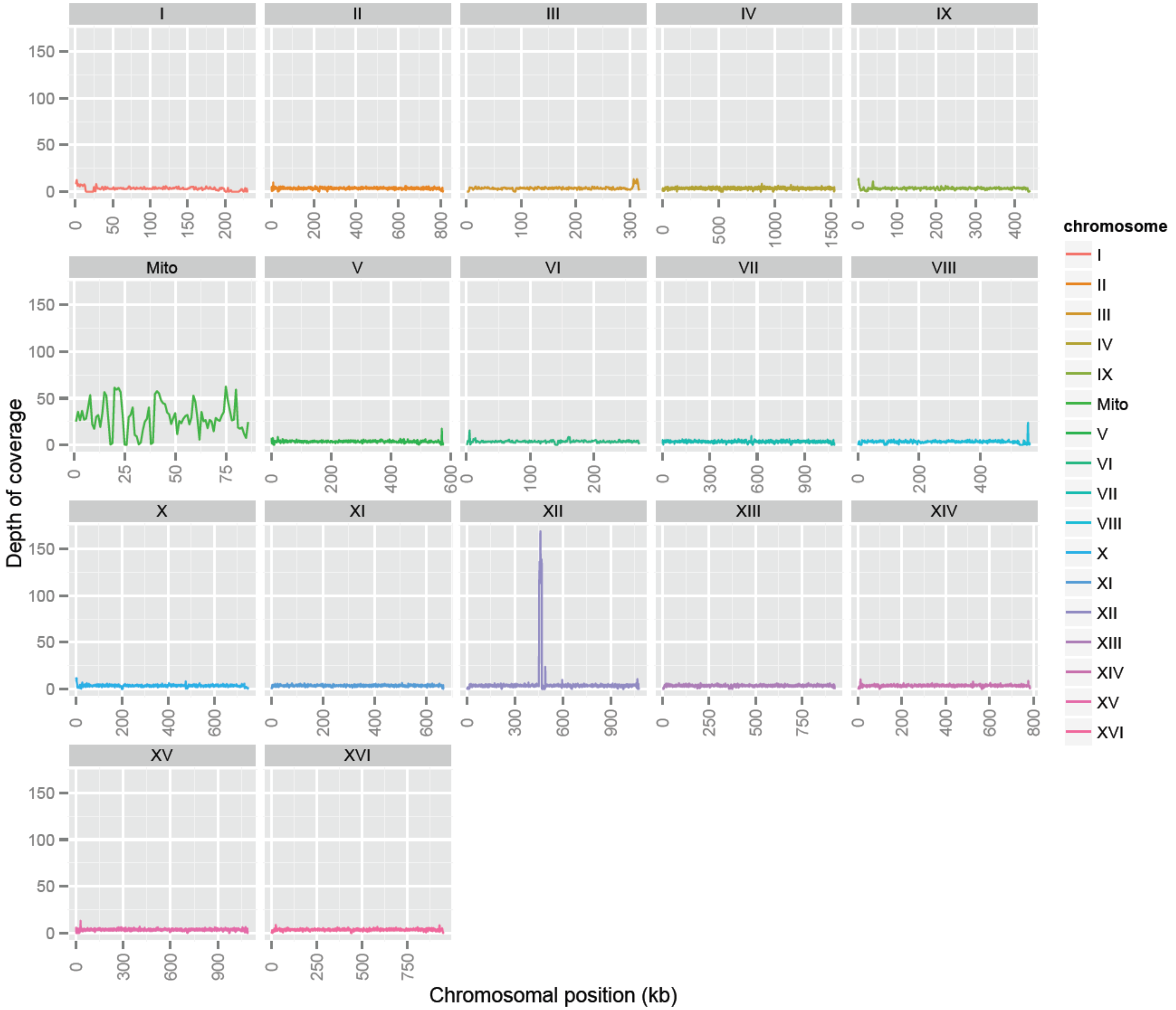
Plot coverage in R

`more mito_coverage` (have a look at the file before you start R)

```
module load r
R
```

```
data <-
read.table("/pico/scratch/userexternal/cbatini0/day3/mito_coverage",
sep="\t", header=T)
names(data)
old_col<-data$Locus
new_col<-gsub("Mito:","",as.character(old_col))
data["pos"]<-new_col
plot(data$pos,data$Depth_for_yeast,type="l")
```

BAM QC - coverage



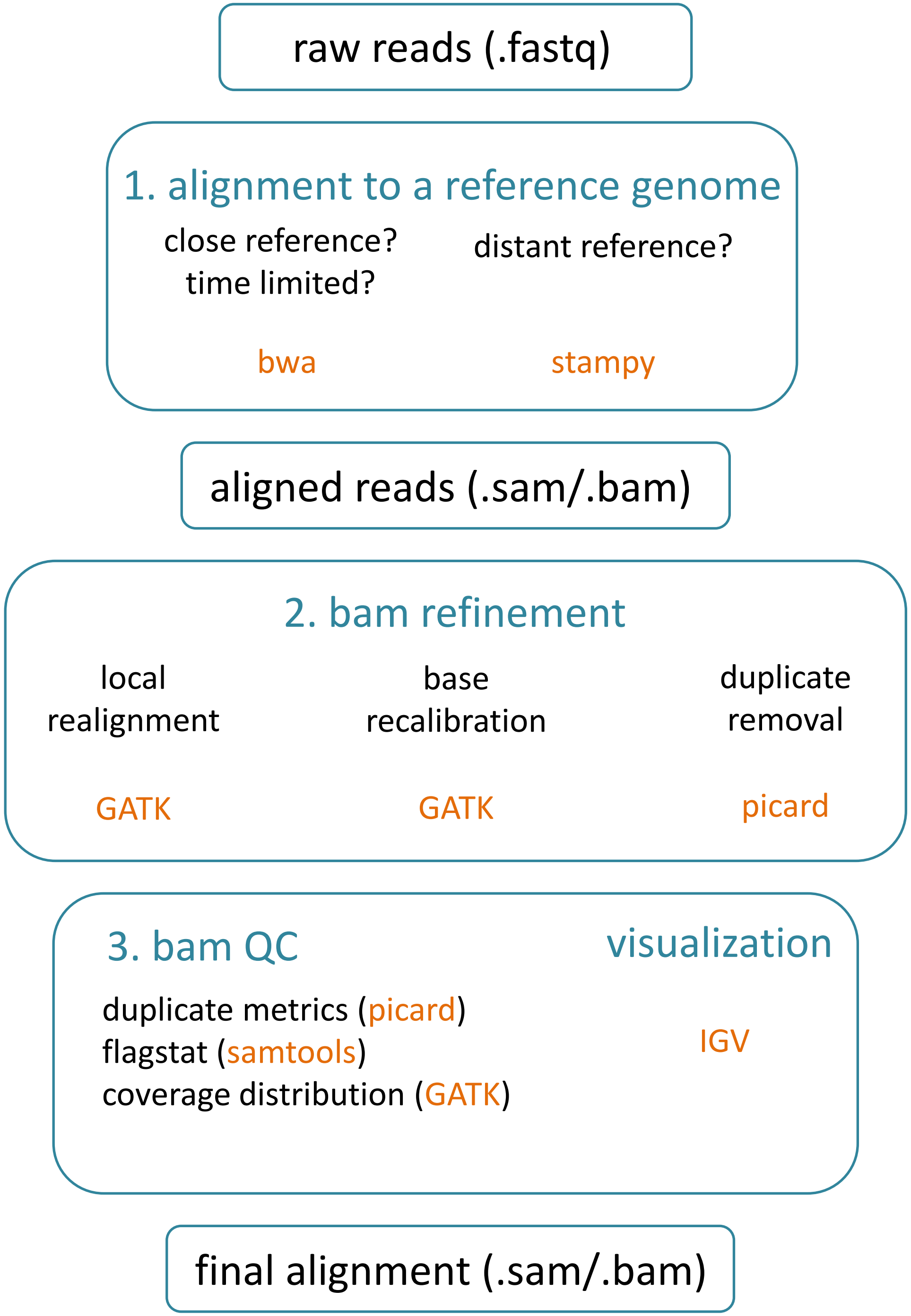
BAM visualization – IGV

extract mtDNA from final BAM

```
samtools view -b -o mito.bam library_final_sorted.bam Mito
```

Launch IGV with Java WebStart

Go to the Download page of IGV, register and launch the 750MB version.



raw reads (.fastq)

1. alignment to a reference genome

close reference?
time limited?

bwa

distant reference?

stampy

aligned reads (.sam/.bam)

2. bam refinement

local realignment

GATK

base recalibration

GATK

duplicate removal

picard

3. bam QC

duplicate metrics (picard)
flagstat (samtools)
coverage distribution (GATK)

visualization

IGV

final alignment (.sam/.bam)